**Problem 1.**

F(n) = F(n-1) + F(n-2)

F (n) = F(n - 1) + F(n -2)

>= 2\*F(n - 2) = 2\*( F( n - 3) + F(n-4) )

>= 2\*2\*F(n-4) = 2^2\*F(n-4)

>= 2^3\*F(n-6)

>= 2^(n/2)\*F(0) = 2^(n/2) > (3/4)^n

**Problem 3.**

* Formulate the recurrence relation. when n = 0

T(n) = 4 if n = 1,T(n) = T(n − 1) + 4 if n > 1

* Calculate value from 1 to n.

T(1) = 4

T(2) = T(1) + 4 = 4 + 4

T(3) = T(2) + 4 = 4 + 4 + 4

T(n) = T(n − 1) + 4 = 4n.

* Prove the formula above.

f(n) = 4n.

f(1) = 4 and f(n) = f(n − 1) + 4. The first part is obviously true.

And f(n) = 4n = 4(n − 1) + 4 = **f(n − 1) + 4**.

* Prove correctness.
  + *Verify valid recursion.* n == 0 || n == 1 is the base case and repeated self- call reduces the input size by 1.
  + *Verify base case outputs are correct*. 0! = 1 and 1! = 1.
  + *Verify inductively that outputs are correct for all n.* recursiveFactorial(j) outputs j! for all j < n, where n > 1 and (n-1)! \* n = n!

**Problem 5.**

a = 1, b = 2, c = 1, d = 1, k = 1 (a < bk)=> The Master Formula T(n) (n).